AIAA Guidance, Navigation and Control Conference and Exhibit
20 - 23 August 2007, Hilton Head, South Carolina

AIAA 2007-6505

Adaptive Neural Network Flight Control Using both Current and Recorded Data

Girish Chowdhary, and Eric N. Johnson†*

Department of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332,

Modern aerospace vehicles are expected to perform beyond their conventional flight envelopes and exhibit the robustness and adaptability to operate in uncertain environments. Augmenting proven lower level control algorithms with adaptive elements that exhibit long term learning could help in achieving better adaptation performance while performing aggressive maneuvers. The current adaptive methodologies which use Neural Network based control methods use only the instantaneous states to tune the adaptive gains. This results in a rank one limitation on the adaptive law. In this paper we propose a novel approach to adaptive control, which uses the current or the online information as well as stored or background information for adaptation. We show that using a combined online and background learning approach it is possible to overcome the rank one limitation on the adaptive law resulting in faster adaptation to the unknown dynamics. Furthermore, we show that using combined online and background learning methods it is possible to guarantee long term learning in the adaptive flight controller, which enhances performance of the controller when it encounters a maneuver that has been performed in the past. We use Lyapunov based methods for showing boundedness of all signals for a proposed method. The performance of the proposed method is evaluated in the high fidelity simulation environment for the GTMAX UAS maintained by the Georgia Tech UAV lab. The simulation results show that the proposed method exhibits long term learning and faster adaptation leading to better performance of the UAS flight controller.

I. Introduction

NEURAL network (NN) controllers have found many successful applications in the Aerospace industry. Neural network based adaptive flight controller for uncertain, nonlinear dynamical systems eliminate the need for offline gain tuning and scheduling methods as well as reduce the money and effort needed to identify and model system dynamics. A neural network can be thought of as a parameterized class of non-linear maps. Multilayer feed-forward neural networks are capable of approximating any continuous unknown nonlinear function or mapping on a compact set[1, 6]. Furthermore Neural Networks have online adaptation capabilities, which can be used to design control laws that can handle uncertainties and nonlinearities in system dynamics and the environment.

Adaptive neural network controllers have been applied to robot arm manipulator control by Lewis, Kim and others[6]. Neural Network controllers are natural choice for Unmanned Aerial Vehicle (UAV) system control due to their capability to adapt to varying dynamics, ease of implementation, and robustness properties. Using Neural Network adaptive flight controllers in UAV control systems design also reduces the effort required in modeling and flight system identification of the flight platform. Calise, Johnson, Kannan and others have implemented Neural Network augmented approximate model inversion controllers with pseudo control hedging successfully for control of various

* Graduate Research Assistant, Girish.Chowdhary@gatech.edu .
† Assosiate Professor, Eric.Johnson@gatech.edu

fixed wing as well as rotary wing UAV systems with significant nonlinearities and saturations in the control loop[2-5]. These controllers have proven successful even in performing highly demanding maneuvers requiring fast adaptation.

The increasing demand on precision, agility, cooperation, and safety in autonomous aerospace systems, their time-variant nature, and the intricacies of real-world operations limit the effectiveness of control architectures employing only lower-level or "steady-state" methodologies. While depending on sound control theory concepts is extremely important; it is clear that true advances can only be made by ensuring intelligence, adaptation, and long term learning in the core control and decision-making architecture. However, the current adaptive laws for neural networks (e.g. 2-7) only use the instantaneous knowledge for adaptation. Hence the current adaptive laws have no real long term memory and hence do not exhibit any improvement in performance when performing maneuvers that have been performed previously. This limitation can also be explained by noting that the rank of the NN weight dynamic is always at most one when only current data is used for NN training. If it could be possible to augment these dynamics to be of a higher rank or even full rank then the additional degrees of freedom could be used to improve the performance of the control system.

In this paper we propose a novel approach to the design of Neural Network adaptive controllers which overcomes the rank-1 limitation and exhibits the properties of semi global learning. We accomplish this by combining current online learning algorithms with a background learning methodology, where the background learning law is a projection of the current learning law into the nullspace of the current learning. We show that one such method which uses Linear in the Parameter NN guarantees boundedness of all signals using a Lyapunov stability approach. Simulation results are analyzed in order to evaluate the performance of the new approach. Our work builds upon the work of the second Author and Seung-Min Oh[8].

II. Neural Network Based Adaptive Control

A brief explanation of a baseline dynamic inversion based online learning Neural Network based adaptive control system is given here. The reader is referred to [2-7] for detailed explanations.

A. Approximate Model Inversion based Adaptive Control

Consider a system of the form:

$$\ddot{x} = f(x, \dot{x}, \delta) \tag{1}$$

Where $x, \dot{x}, \delta \in \Re^n$. We introduce a pseudo control input ν which represents a desired $\ddot{x}$ and is expected to be approximately achieved by the actuating signal δ, in the following manner:

$$\ddot{x} = \nu \tag{2}$$

Where,

$$\nu = f(x, \dot{x}, \delta) \tag{3}$$

In a model inversion scheme the actual control input δ is found by inverting Eq.(3). However since the function $f(x, \dot{x}, \delta)$ is usually not exactly known or hard to invert, an approximation is introduced as:

$$\nu = \hat{f}(x, \dot{x}, \delta). \tag{4}$$

Based on the approximation above the actuator command is determined by an approximate dynamic inversion of the form

$$\delta_{cmd} = \hat{f}^{-1}(x, \dot{x}, \nu). \tag{5}$$

This results in a modeling error in the system dynamics,

$$\ddot{x} = \nu - \Delta(x, \dot{x}, \delta) \tag{6}$$

Where,

$$\Delta(x,\dot{x},\delta) = f(x,\dot{x},\delta) - \hat{f}(x,\dot{x},\delta) \tag{7}$$

The approximation, $\hat{f}$ is chosen such that an inverse with respect to δ exists. Figure 1depicts a more specific form of an approximate dynamic inversion-based Neural Network adaptive controller including actuator a PCH compensation.

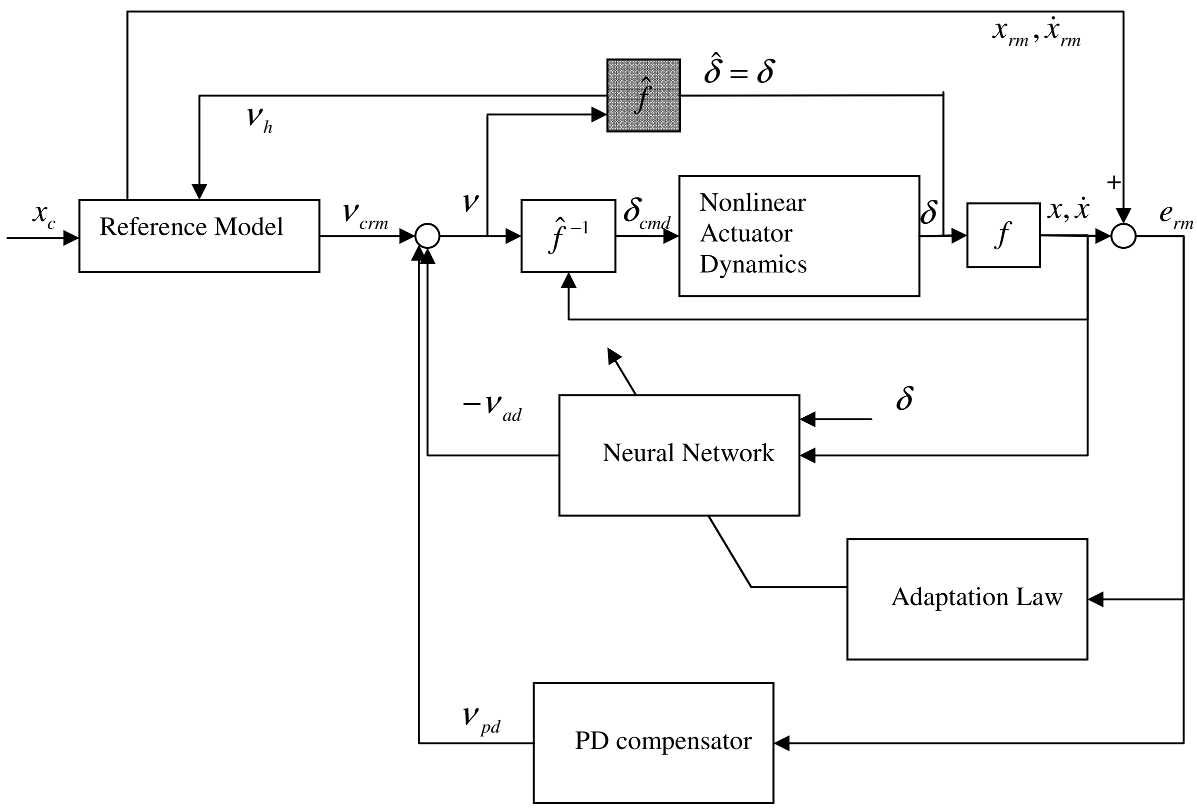

Figure 1 Neural Network Adaptive Control using Approximate Model Inversion and PCH compensation

Based on approximation in Eq.(4), the actuator command is determined by an approximate dynamic inversion of the form

$$\delta_{cmd} = \hat{f}^{-1}(x,\dot{x},\delta) \tag{8}$$

Where ν is termed the 'pseudo-control', and represents a desired $\ddot{x}$ that is expected to be approximately achieved by δ_{cmd}. This dynamic inversion assumes perfect actuator dynamics and hence does not take into account effects such as actuator saturation, or rate limitation. As a result the actual command may not equal the achieved command due to the characteristics of the actuator (which may further vary with time). Incorporating the actuator dynamics in the actual nonlinear inversion presents other difficulties arising due to various discontinuous actuator characteristics, such as actuator saturation, discrete (quantized) control, rate limitation, time delays, and unmodelled dynamics. The Neural Network element will attempt to adapt to these characteristics even when it might not be desirable to do so. Pseudo Control Hedging (PCH)[4] is one method that can handle this problem. This method prevents the adaptive elements of the adaptive control system from trying to adapt to a class of unwanted plant input characteristics.

The pseudo-control hedge signal (ν_h) is defined as the difference between the commanded pseudo-control input and the actually achieved pseudo-control input. This difference is computed by using an estimated actuator position based on a model or measurement. This estimate is then used to get the pseudo-control hedge as the difference between commanded pseudo-control and the estimated actual pseudo-control.

$$\nu_h = \hat{f}(x,\dot{x},\delta_{cmd}) - \hat{f}(x,\dot{x},\hat{\delta}) = \nu - \hat{\nu} \tag{9}$$

Figure 1 illustrates the manner in which pseudo control hedging can be achieved for a position and rate limited actuator. The PCH signal is introduced as an addition input into the reference model, forcing it to 'move back'. Hence the reference model dynamics with PCH become:

$$\ddot{x}_{rm} = \nu_{crm}(x_{rm},\dot{x}_{rm},x_c,\dot{x}_c) - \nu_h \tag{10}$$

Where $x_c, \dot{x}_c$ represent external commands. The instantaneous pseudo-control output of the reference model in the feed-forward path is not changed by the use of PCH and is ν_{crm}.

$$\nu_{crm} = f_{rm}(x_{rm},\dot{x}_{rm},x_c,\dot{x}_c) \tag{11}$$

B. Model Tracking Error Dynamics

The total pseudo-control signal for the system is now constructed by the three components:

$$\nu = \nu_{crm} + \nu_{pd} - \nu_{ad} \tag{12}$$

Where ν_{crm} is the pseudo-control signal generated by the reference model in Eq. (11), ν_{pd} is the output of a linear compensator, and ν_{ad} is the Neural Network adaptation signal. The linear compensator (ν_{pd}) can be designed using standard linear control design techniques which render the closed loop system stable, these include P-D(Proportional-Derivative) compensation or LQR (Linear Quadratic Regulator) compensation. For the second order system PD compensation is expressed by

$$\nu_{pd} = [K_p \quad K_d]e \tag{13}$$

Where the reference model tracking error is defined as:

$$e = \begin{bmatrix} x_{rm} - x \\ \dot{x}_{rm} - \dot{x} \end{bmatrix} \tag{14}$$

The model tracking error dynamics are found by differentiating e:

$$\dot{e} = Ae + B\left[\nu_{ad}(x,\dot{x},\delta) - f(x,\dot{x},\delta) + \hat{f}(x,\dot{x},\delta)\right] \tag{15}$$

Where,

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix}, B = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{16}$$

Where both K_p and K_d are real positive matrices. With the above form, A is Hurwitz. When one assumes that the plant inputs δ are exactly known then the error dynamics can be represented as:

$$\begin{aligned} &\dot{e} = Ae + B[\nu_{ad}(x,\dot{x},\delta) - \Delta(x,\dot{x},\delta)] \\ &where, \\ &\Delta(x,\dot{x},\delta) = f(x,\dot{x},\delta) - \hat{f}(x,\dot{x},\delta) \end{aligned} \tag{17}$$

Is regarded as the model error to be approximated and cancelled by ν_{ad}, the output of the Neural Network. We define the signal r as:

$$r = e^T PB \qquad \in \Re^{n_3 \times 1} \tag{18}$$

Where $P \in \Re^{2n\times 2n}$ is the positive definite solution to the Lyapunov equation:

$$A^T P + PA + Q = 0 \tag{19}$$

C. Neural Network Based Adaptation

Single Hidden Layer (SHL) Perceptron NNs are universal approximators. They can approximate any smooth nonlinear function to within arbitrary accuracy given sufficient number of hidden layer neurons and input information[20]. The input output map of the SHL NN can be expressed in compact matrix form as:

$$v_{ad}(W,V,\bar{x}) = W^T \sigma(V^T \bar{x}) \qquad \in \Re^{n_3\times 1} \tag{20}$$

Where the following definitions are used:

$$\bar{x} = \begin{bmatrix} b_v \\ x_{in} \end{bmatrix} = \begin{bmatrix} b_v \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \qquad \in \Re^{(n_1+1)\times 1} \tag{21}$$

$$\sigma(z) = \begin{bmatrix} b_{vw} \\ \sigma_1(z_1) \\ \sigma_2(z_2) \\ \cdot \\ \cdot \\ \sigma_n(z_n) \end{bmatrix} \qquad \in \Re^{(n_2+1)\times 1} \tag{22}$$

$$V = \begin{bmatrix} \theta_{1,1} & \cdots & \theta_{1,n2} \\ v_{1,1} & \cdots & \theta_{1,n2} \\ \vdots & \ddots & \vdots \\ v_{n1,1} & \cdots & \theta_{n1,n2} \end{bmatrix} \in \Re^{(n_1+1)\times n_2} \tag{23}$$

$$W = \begin{bmatrix} \theta_{1,1} & \cdots & \theta_{1,n2} \\ w_{1,1} & \cdots & w_{1,n2} \\ \vdots & \ddots & \vdots \\ w_{n2,1} & \cdots & w_{n2,n3} \end{bmatrix} \in \Re^{(n_2+1)\times n_3} \tag{24}$$

Where, $\bar{x}$ is the input vector, σ is the sigmoidal activation function vector, V is an input layer to hidden layer weight matrix, W is a hidden layer to output layer weight matrix, and v_{ad} is the NN output. $b_v \geq 0$ and $b_w \geq 0$ are input biases that allow the thresholds θ_v and θ_w to be included in the weight matrix V and W. n_1, n_2, and n_3 represent the number of input, hidden, and output layer nodes respectively.

Input to hidden layer neuron is:

$$z = V^T \bar{x} = \begin{bmatrix} z_1 \\ \vdots \\ \vdots \\ z_{n2} \end{bmatrix} \in \Re^{n_2 \times 1} \tag{25}$$

The sigmoidal activation function used is:

$$\sigma_j(z_j) = \frac{1}{1 + e^{-a_j z_j}} \tag{26}$$

Details on Neural Network theory can be found in reference [2,6,7].

III. Online Learning NN Adaptive Control and Rank-1 Limitation

Neural Networks are considered to be excellent function approximators, that is, they can approximate any smooth nonlinear function within a compact set to arbitrary accuracy given enough number of input layer neurons and proper inputs. We present a brief proof for the standard backpropogation method of NN online weight adaptation.

The following online adaptive law guarantees the boundedness of all signals[6, 2]

$$\begin{aligned} \dot{W} &= -\sigma r \Gamma_w \\ \dot{V} &= -\Gamma_v \bar{x} r W^T \sigma' \end{aligned} \tag{27}$$

Consider a positive definite Lyapunov candidate of the form:

$$L(e, \tilde{W}, \tilde{V}) = \frac{1}{2} e^T P e + \frac{1}{2} tr\left\{\left(\tilde{W} \Gamma_w^{-1} \tilde{W}^T\right)\right\} + \frac{1}{2} tr\left\{\left(\tilde{V} \Gamma_v^{-1} \tilde{V}^T\right)\right\} \tag{28}$$

Where, $\tilde{W} = W - W^*$ and $\tilde{V} = V - V^*$, where W^* and V^* denote the ideal weights for the NN. Note that:

$$\begin{aligned} &L(e, \tilde{W}, \tilde{V}) = 0 \quad iff \quad e = 0, e_i = 0, W = W^* \; and \quad V = V^* \\ &and, \quad L(e, \tilde{W}, \tilde{V}) > 0 \quad otherwise \\ &furthermore, \quad L(e, \tilde{W}, \tilde{V}) \to \infty \quad as \quad e, \tilde{W}, \tilde{V} \to \infty \end{aligned} \tag{29}$$

Hence the Lyapunov candidate is radially unbounded[10,11]. Taking the time derivative of the Lyapunov candidate we have,

$$\dot{L}(e, \tilde{W}, \tilde{V}, \sigma) = -\frac{1}{2} e^T Q e + r(\nu_{ad} - \Delta) + tr\left\{\left(\dot{W} \Gamma_w^{-1} \tilde{W}^T\right)\right\} + tr\left\{\left(\tilde{V} \Gamma_v^{-1} \dot{V}^T\right)\right\} \tag{30}$$

Expanding the NN model cancellation error[6,6] we have,

$$\nu_{ad} - \Delta = W^T \sigma(V^T \bar{x}) - W^{*T} \sigma(V^{*T} \bar{x}) = \tilde{W}^T \sigma\left(V^T \bar{x}\right) + W^T \sigma'\left(V^T \bar{x}\right) \tilde{V}^T \bar{x} + \text{H.O.T.} \tag{31}$$

<u>Assumption</u>: In the above equation H.O.T. represents higher order terms which we omit for the sake of clarity. It is possible to obtain bounds on the H.O.T. which changes the adaptation law accordingly[3,4].

Then,

$$\dot{L}(e, \tilde{W}, \tilde{V}) = -\frac{1}{2} e^T Q e + tr\left\{\left(\dot{W} \Gamma_w^{-1} + \sigma r\right) \tilde{W}^T\right\} + tr\left\{\tilde{V}^T \left(\Gamma_v^{-1} \dot{V}^T + \bar{x} r \overline{W}^T \sigma'\right)\right\} \tag{32}$$

By setting:

$$\dot{W}\Gamma_w^{-1} + \sigma r = 0 \tag{33}$$

$$\Gamma_v^{-1}\dot{V}^T + \bar{x}r\overline{W}^T\sigma' = 0 \tag{34}$$

We have,

$$\dot{L}(e,\tilde{W},\tilde{V}) = -\frac{1}{2}e^T Qe < 0 \tag{35}$$

Establishing Lyapunov stability, we note that equation 35 can be written as a strict inequality based on assumption 1. However, if the H.O.T. terms of equation 31 are considered equation 35 does not result in a strict inequality, and the LaSalle Theorem and the Barbalat's lemma[10] needs to be used for ascertaining Lyapunov stability.

Solving Eq. (32) and Eq(33) yields the adaptive laws,

$$\begin{aligned}\dot{W} &= -\sigma r\Gamma_w \\ \dot{V} &= -\Gamma_v\bar{x}rW^T\sigma'\end{aligned} \tag{36}$$

Fact 1: Every matrix of *rank one* has the simple form A=uv^T Where A is $n\times m$ matrix, u is $n\times 1$ vector and v is $m\times 1$ vector.

Using the above fact from linear algebra it is easy to see that since $\sigma\in\Re^{(n_2+1)\times 1}$ and $r\Gamma_w\in\Re^{(1\times n_3)}$ then $\dot{W}$ is always at most a *rank one* matrix. Similarly $\dot{V}$ is also at most *rank one* because $\Gamma_v\bar{x}\in\Re^{(n_1+1)\times 1}$ and $rW^T\sigma'\in\Re^{(1\times n_2)}$. Hence, even though the NN weight adaptation matrices have a matrix form, their rank is always at most one. This may affect the performance of the NN law. In order to overcome the *rank one* limitation it is proposed to utilize online as well as background learning by using current as well as stored data in the NN weight adaptation process. In this paper we show that using this approach yields better performance since it makes use of all the information available for the adaptation purposes. We also show that use of current as well as stored date improves global learning behavior and guarantees long term learning of the adaptation.

IV. Combined Online and Background Learning Adaptive Control

A. Choice of the background learning law

Any learning that does not immediately affect the instantaneous learning (that is, does not directly affect ν_{ad}) has the following form:

$$\dot{W}_b^T\sigma = 0 \tag{37}$$

$$\dot{V}_b^T\Gamma_v\bar{x} = 0 \tag{38}$$

where the subscript b denotes the background learning law. This condition ensures that the background learning adaptation law is orthogonal to the underlying vector space of the instantaneous learning. This indicates that the orthogonal projection of any NN learning law can be used as a background learning law. In this paper we consider the orthogonal projection of the learning law for the W and the V matrix onto the orthogonal subspace of the span of equation 37 and equation 38 in the following form

$$\dot{W}_b = \left(I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\right)\dot{W}_t \tag{39}$$

$$\dot{V}_b = \left(I - \frac{\Gamma_V \bar{x}\bar{x}^T \Gamma_V}{\bar{x}^T \Gamma_V \Gamma_V \bar{x}} \right) \dot{V}_t \tag{40}$$

where the subscript t denotes any suitable NN learning law. It is to be noted that equation 39 and equation 40 are also the optimal solutions to the Lagrange's constrained minimization method by minimizing $\left\| \dot{W}_t - \dot{W}_b \right\| + \left\| \dot{V}_t - \dot{V}_b \right\|$ corresponding to the Frobenius norm[4].

One reasonable choice for the training of the background learning law is to train the NN using stored data along with current data in order to improve global learning behavior of the NN and guarantee long term adaptation. In the proposed method this is achieved by using current data as well as a stored 'history stack'[8]. Both data are used concurrently in the adaptation process.

The combined control law then has the form:

$$\dot{W} = \dot{W}_t + \left(I - \frac{\sigma\sigma^T}{\sigma^T \sigma} \right) \sum_{i=1}^{p} W_{t_i} \tag{42}$$

And

$$\dot{V} = \dot{V}_t + \left(I - \frac{\Gamma_V \bar{x}\bar{x}^T \Gamma_V}{\bar{x}^T \Gamma_V \Gamma_V \bar{x}} \right) \sum_{i=1}^{p} \dot{V}_t \tag{43}$$

B. Selection of data points for background learning

Selection of NN inputs for background learning is not a trivial problem, since these inputs impact the global learning properties of the combined online and background learning approach. Detailed discussion on some methods of selecting data points can be found in [8], we suffice here by mentioning that we select data points that satisfy the following criterion:

$$\frac{\left(\bar{x} - \bar{x}_p \right)^T \left(\bar{x} - \bar{x}_p \right)}{\bar{x}^T \bar{x}} > \varepsilon_{\bar{x}} \tag{44}$$

Here the subscript p denotes the index of the last data point stored. The above method ascertains that only those data points are selected that are sufficiently different from the last data point stored. Once the data points are selected, the model error relating to that data point must be observed and stored. We achieve this by using an online implementation of optimal fixed point smoothing[12]. In the given framework of adaptive control the model error Δ_i for the i[th] data point is

$$\Delta_i (x_i, \dot{x}_i, \delta_i) = f_i (x_i, \dot{x}_i, \delta_i) - \hat{f} (x_i, \dot{x}_i, \delta_i) \tag{45}$$

Using equation 5, the above can be expressed as:

$$\Delta_i (x_i, \dot{x}_i, \delta_i) = \ddot{x}_i - \nu_i \,. \tag{46}$$

Once a point is selected for storing, the fixed point smoothing algorithm is initiated until a sufficiently accurate estimate of $\ddot{x}_i$ is obtained. Using this estimate and stored values of ν_i an estimate of the model error for the i[th] data point is obtained. The residual signal that is used in the background learning adaptation is:

$$r_i = W^T \sigma(V^T \bar{x}) - \Delta_i \,. \tag{47}$$

Considering equation 16, it is seen that the residual signal in this form for the background learning NN attempts to reduce is the difference between the current estimate of the model error and a stored estimate of the model error. Hence, one can say that by using this method, the background learning NN attempts to adapt the W and V matrices of the NN in such a way that the model error for multiple data points is simultaneously reduced.

Alternatively, the residual signal r can be formed by simulating the error dynamics[8] by integrating equation 17 for the i^{th} data point and then using equation 18.

C. Approximate model inversion adaptive control using Combined instantaneous and background learning NN

We now present a novel NN weight training law that uses both current and stored data for a Linear In the Parameters (LIP) NN.

A LIP has the simple form given by:

$$\nu_{ad}(W,V,\bar{x}) = W^T\sigma(\bar{x}) \qquad \in \Re^{n3\times 1} \tag{48}$$

Where σ is an appropriate basis function. The benefit of using a LIP NN is that only the W matrix of equation 24 containing the weights of the NN needs to be tuned.

Theorem 1*: Consider the system in equation 1 and the inverting controller of equation 5, the following combined online and background adaptive laws for a LIP NN of equation 48 guarantee the boundedness of all signals:*

$$\dot{W} = -\sigma r\Gamma_w - \left(kI + I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\right)\sum_{i=1}^{p}\sigma(\bar{x}_i)r_i\Gamma_W \tag{49}$$

With the sigmoidal activation function defined as:

$$\hat{\sigma} = \left(kI + \left(I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\right)\right)\begin{bmatrix} b_w \\ \sigma_1(\bar{x}_{n_1}) \\ \dots \\ \sigma_{n_2}(\bar{x}_{n_2}) \end{bmatrix} \qquad \in \Re^{(n_2+1)\times 1} \tag{50}$$

Where σ_i are found from equation 26 with $z = \bar{x}$, and k is a predefined, nonzero constant.

<u>**Proof**</u>: The orthogonality of the operators in equation 39 can be expressed as,

$$\left(I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\right)\sigma = 0 \tag{51}$$

Also note that,

$$\nu_{ad} - \Delta = W^T\hat{\sigma}(\bar{x}) - W^{*T}\hat{\sigma}(\bar{x}) = \tilde{W}^T\left(kI + \left(I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\right)\right)\sigma(\bar{x}) = \tilde{W}^T k\sigma(\bar{x}). \tag{52}$$

And,

$$\nu_{ad_i} - \Delta_i = W^T\hat{\sigma}(\bar{x}_i) - W^{*T}\hat{\sigma}(\bar{x}_i) = \tilde{W}^T\left(kI + \left(I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\right)\right)\sigma(\bar{x}). \tag{53}$$

Where W^*, V^* denote the ideal NN weights. The error dynamics for the ith data point can be written as:

$$\dot{e}_i = Ae_i + B\lfloor \nu_{ad_i}(x,\dot{x},\delta) - \Delta_i(x,\dot{x},\delta) \rfloor \tag{54}$$

Consider a Lyapunov candidate of the form:

$$L(e,\tilde{W},\tilde{V},\sigma) = \frac{1}{2}e^T Pe + \frac{1}{2}\sum_{i=1}^{p} e_i^T Pe_i + \frac{1}{2}tr\left\{\left(\tilde{W}_b \Gamma_w^{-1} \tilde{W}_b{}^T\right)\right\} + \frac{1}{2}tr\left\{\left(\tilde{W}_t \left(\frac{\Gamma_W}{k}\right)^{-1} \tilde{W}_t{}^T\right)\right\} \tag{55}$$

$$\begin{aligned} &L(e,e_i,\tilde{W}) = 0 \quad && iff \quad e = 0, e_i = 0, W = W^* \; and \\ &and, \quad L(e,e_i,\tilde{W}_t,\tilde{W}_b) > 0 \quad && otherwise \\ &furthermore, \quad L(e,e_i,\tilde{W}_t,\tilde{W}_b) \to \infty \quad && as \quad e, e_i, \tilde{W}_t, \tilde{W}_b \to \infty \end{aligned} \tag{56}$$

The last condition is the 'radially unbounded' condition[10,11]. Taking time derivative of the Lyapunov candidate along the trajectory of system described by equation 54 we have:

$$\begin{aligned} \dot{L}(e,\tilde{W},\tilde{V},\sigma) = &-\frac{1}{2}e^T Qe + r^T(\nu_{ad} - \Delta) - \frac{1}{2}\sum_{i=1}^{p} e_i^T Qe_i + \sum_{i=1}^{p} r_i^T\left(\nu_{ad^i\,i} - \Delta_i\right) + tr\left\{\left(\dot{W}_t\left(\frac{\Gamma_W}{k}\right)^{-1} \tilde{W}_t{}^T\right)\right\}.... \\ &+ .tr\left\{\left(\dot{W}_b \Gamma_w^{-1} \tilde{W}_b{}^T\right)\right\} \end{aligned} \tag{57}$$

Using equation 52 and equation 53 the above equation can be written as:

$$\dot{L}(e,\tilde{W},\tilde{V},\sigma) = -\frac{1}{2}e^T Qe - \sum_{i=1}^{p}\frac{1}{2}e_i{}^T Qe_i + tr\left\{\left(\dot{W}_t\left(\frac{\Gamma_W}{k}\right)^{-1} + k\sigma r^T\right)\tilde{W}^T\right\} + tr\left\{\left(\dot{W}_b \Gamma_w^{-1} + \left(kI + I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\right)\sum_{i=1}^{p}\sigma r^T\right)\tilde{W}^T\right\} \tag{58}$$

Then by setting,

$$\dot{W}_t\left(\frac{\Gamma_W}{k}\right)^{-1} + k\sigma r^T = 0 \tag{59}$$

And

$$\dot{W}_b \Gamma_w^{-1} + \left(kI + I - \frac{\sigma\sigma^T}{\sigma^T\sigma}\right)\sum_{i=1}^{p}\sigma r^T = 0 \tag{60}$$

And noting that

$$\dot{W} = \dot{W}_t + \dot{W}_b \tag{61}$$

We arrive at the update law given in equation 49. Furthermore, the time derivative of the Lyapunov candidate reduces to:

$$\dot{L}(e,\tilde{W},\tilde{V},\sigma) = -\frac{1}{2}e^T Qe - \sum_{i=1}^{p}\frac{1}{2}e_i{}^T Qe_i < 0 \qquad e \in \Re^{2n} \quad e \neq 0 \tag{62}$$

Since, $L > 0$ and $\dot{L} < 0$, the NN adaptive law given in equation 49 guarantees boundedness of all signals based on the Lyapunov approach for the control system of equation 5. □

Remark 1:

1. When a data point is added the discrete change in the Lyapunov function is zero.
2. When a data point is dropped the net change in the Lyapunov function is negative.
3. Due to 1 and 2 the system signals are all bounded in the sense of Lyapunov stability.

Also note that since the nonzero constant *k* can be arbitrarily chosen, it is possible to choose small enough k such that equation 60 forms an arbitrarily close approximation of equation 42 with the adaptive law of equation 59.

Method 2*: Consider the system in equation 1 and the inverting controller of equation 5, the following combined online and background adaptive laws for the SHL NN of equation 20 are proposed:*

$$\dot{W} = -\sigma r \Gamma_w - \left(I - \frac{\sigma\sigma^T}{\sigma^T\sigma} \right) \sum_{i=1}^{p} \sigma(V^T \bar{x}_i) r_i \Gamma_W \tag{63}$$

$$\dot{V} = -\Gamma_V \bar{x} r W^T \sigma' - \left(I - \frac{\Gamma_V \bar{x}\bar{x}^T \Gamma_V}{\bar{x}^T \Gamma_V \Gamma_V \bar{x}} \right) \sum_{i=1}^{p} \Gamma_V \bar{x}_i r_i W^T \sigma'(V^T \bar{x}_i)\,. \tag{64}$$

V. Demonstration of concept for the adaptive control of an inverted pendulum

To illustrate the concept of background learning augmented adaptive control, we present a simple example with a low dimensional problem.

The inverted pendulum system described by the following equation is to be controlled:

$$\ddot{x} = \delta + \sin(x) - |\dot{x}|\dot{x}\,. \tag{65}$$

Where δ the actuator model, x describes the position of the pendulum, the last two terms are regarded as unknown and represent a significant model error. We assume that a measurement for $\ddot{x}$ is not available and that the system outputs are corrupted by Gaussian white noise. Consequently, an optimal fixed lag smoother is used to estimate the model error of equation 6 for points sufficiently far in the past. The reference model desired dynamics are that of a second order system. We use a *history stack* of 5 data points, replacing the oldest data point as newer points are selected. Background learning point selection is based on equation 44. The background learning method used is that of *Theorem 1*.

Figure 2 shows the performance of the NN based adaptive controller for the plant in Eq. 65. Square waves are commanded at regular intervals. No considerable improvement is seen over the span of the input command. This indicates that the adaptive control has no long term memory and does not show better performance when presented with a task that it has encountered before. Figure 3 shows the history of the NN weight adaptation, the forgetting nature of the adaptive law is clearly seen.

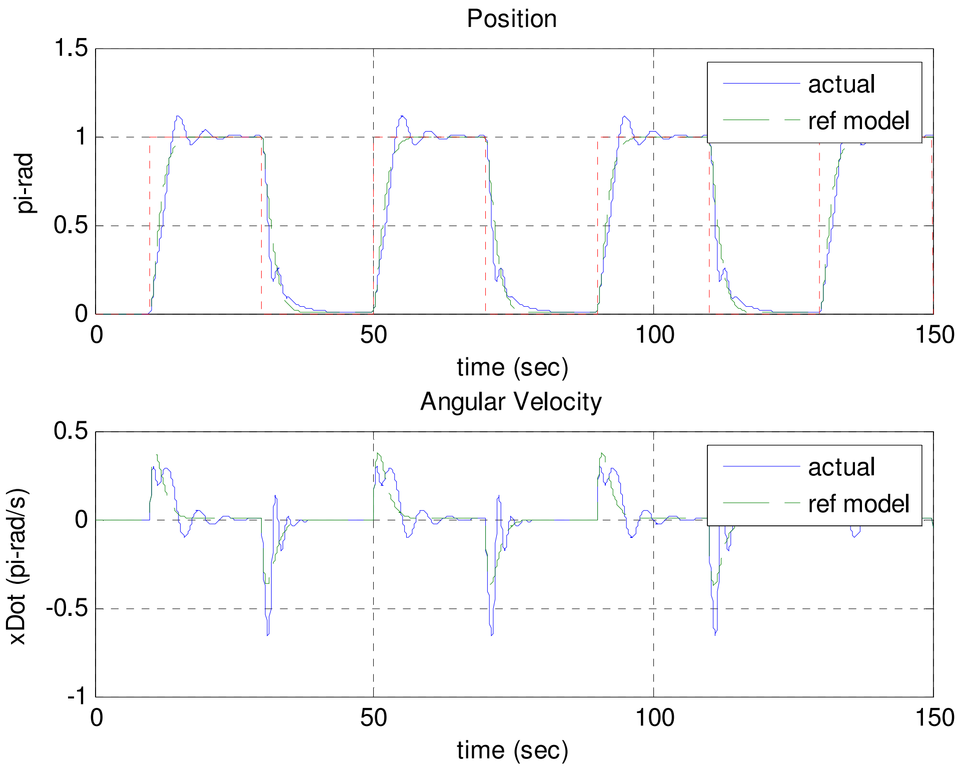

Figure 2 Comparison of states, only online adaptation

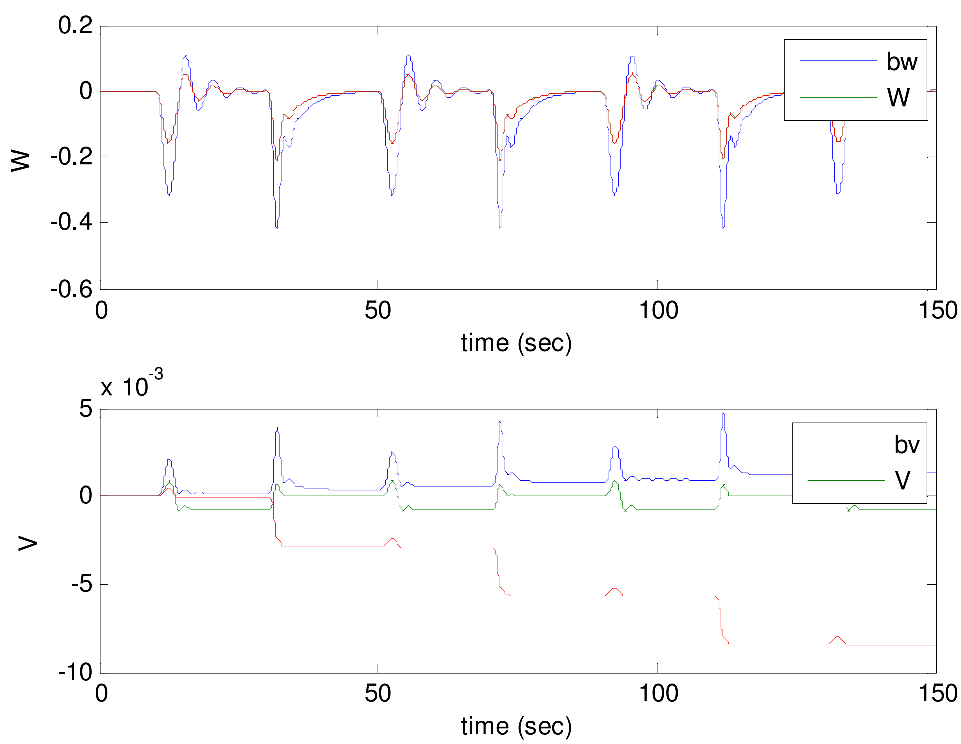

Figure 3 NN weight adaptation W and V, only online adaptation

Figure 4 shows the state comparison when background learning is used, while Figure 5 explicitly shows the evolution of both position and angular velocity error. It is clearly seen that the controller performance improves through subsequently repeated commands, which exhibits long term learning in the adaptive element. To further characterize the impact of background learning we consider the following two criterions:

1. Comparatively quicker convergence of NN weights to constant values (Figure 6). This behavior indicates that the NN is able to adapt to the unknown model error faster when background learning is used.
2. Convergence of equation 47 for each stored data point (Figure 7). When background learning is on, the difference between the stored estimate of model error and the current estimate of model error reduces with time. This indicates that the NN is concurrently adapting to various data points, exhibiting semi-global learning.

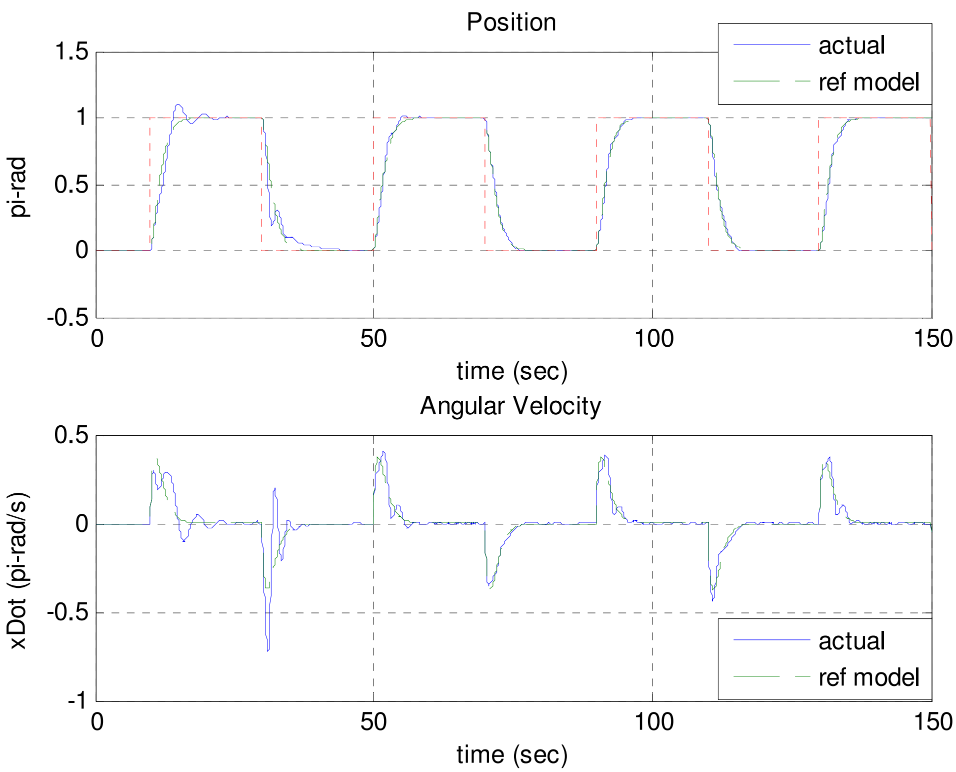

Figure 4 Comparison of states, with background learning method 1

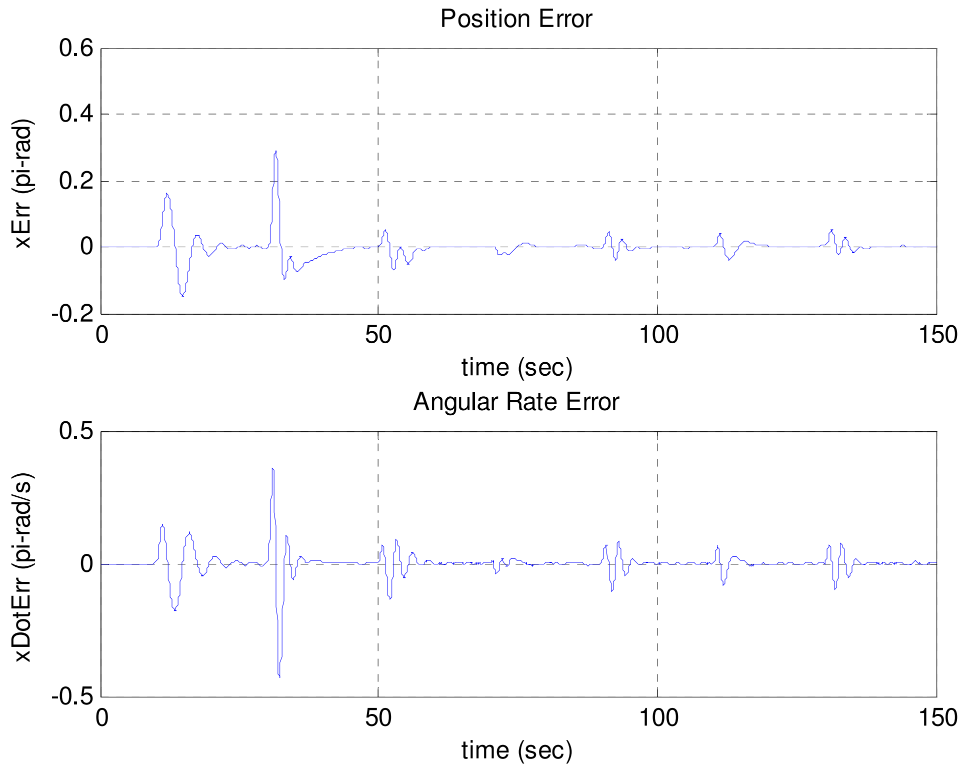

Figure 5 Position and angular rate error with background learning adaptive controller

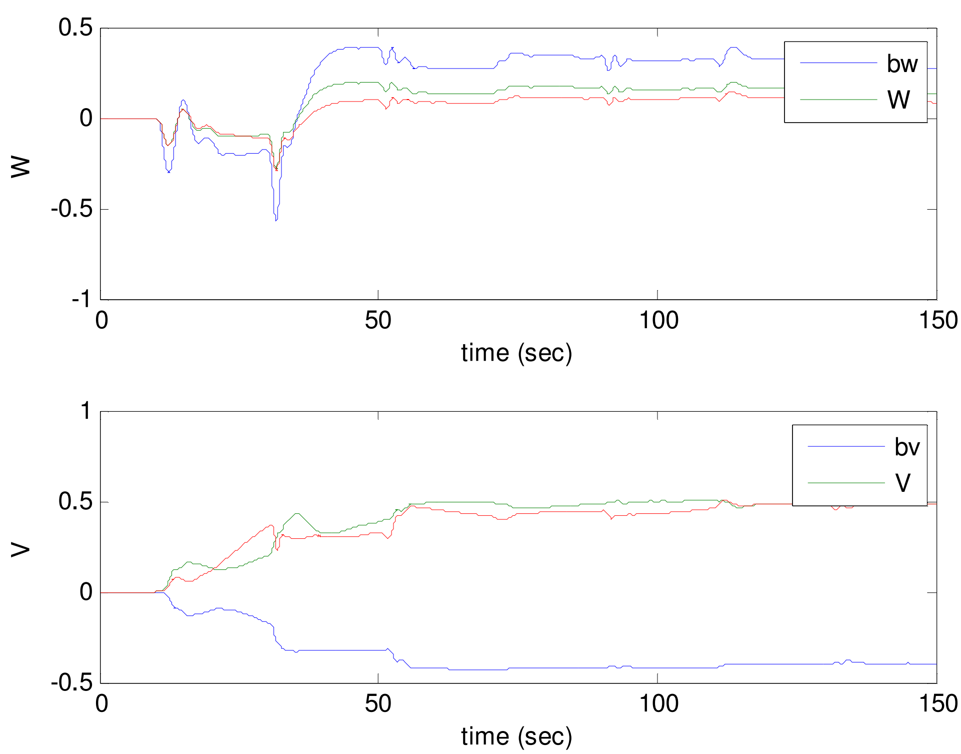

Figure 6: NN adaptation weights, W and V, with combined online and background learning, method 1

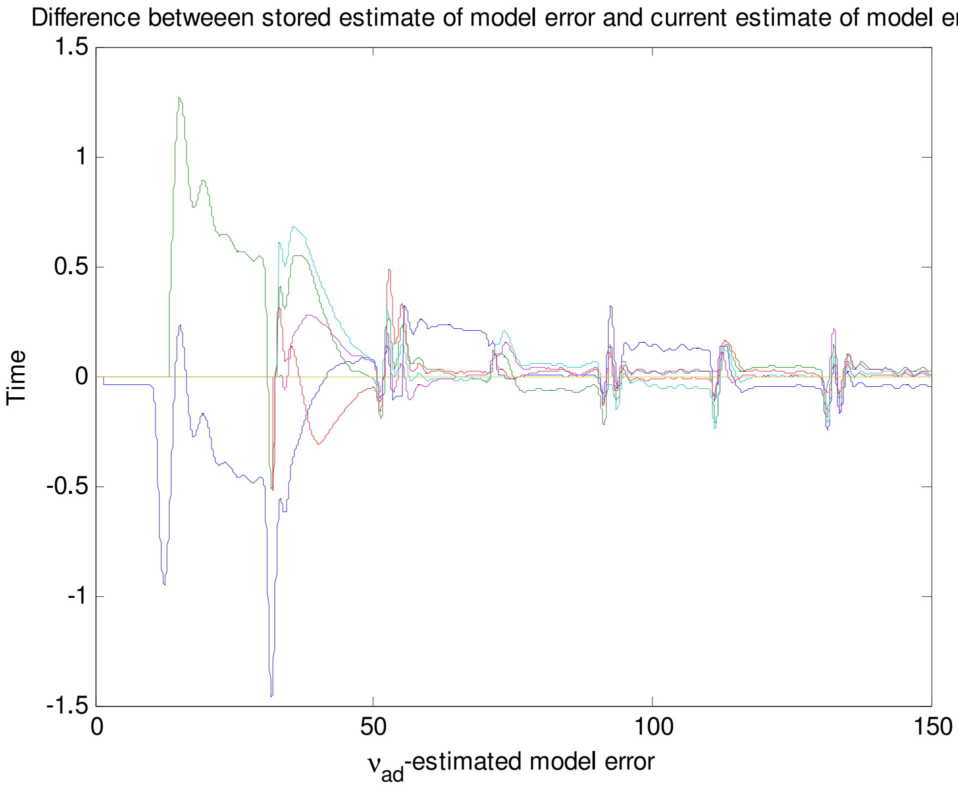

Figure 7 Difference between stored estimate of model error and current estimate of model error with background learning on

VI. Implementation on a high fidelity flight simulator

The Georgia Tech UAV lab maintains a high fidelity Software In the Loop flight simulator, complete with sensor emulation, detailed actuator models, disturbance simulation, and a high fidelity dynamical model. Our target platform is the Georgia Tech GTMAX Unmanned Aerial System (UAS), which is based on the versatile YAMAHA RMAX helicopter (Figure 8). The following results have been simulated on the GTMAX SITL simulation. Since this is a higher dimensional problem, theory indicates that the impact of background learning should be more significant.

Figure 8 The Georgia Tech GTMAX, in landing auto approach

The GTMAX uses an approximate model inversion adaptive controller characterized equivalently to the description in section II, a detailed description can be found in reference 2 and reference 3.

We command four successive forward step inputs with arbitrary delay between any two successive steps. The performance of the inner loop controller is characterized by the errors in the three body angular rates (namely roll rate *p*, pitch rate *q* and yaw rate *r*). As the rotorcraft accelerates and decelerates in forward step inputs the body roll rate *q* dominates. Figure 9 shows the performance of the inner loop controller with only instantaneous adaptation in the NN. It is clearly seen that there is no considerable improvement in the roll rate error as the controller follows successive step inputs.

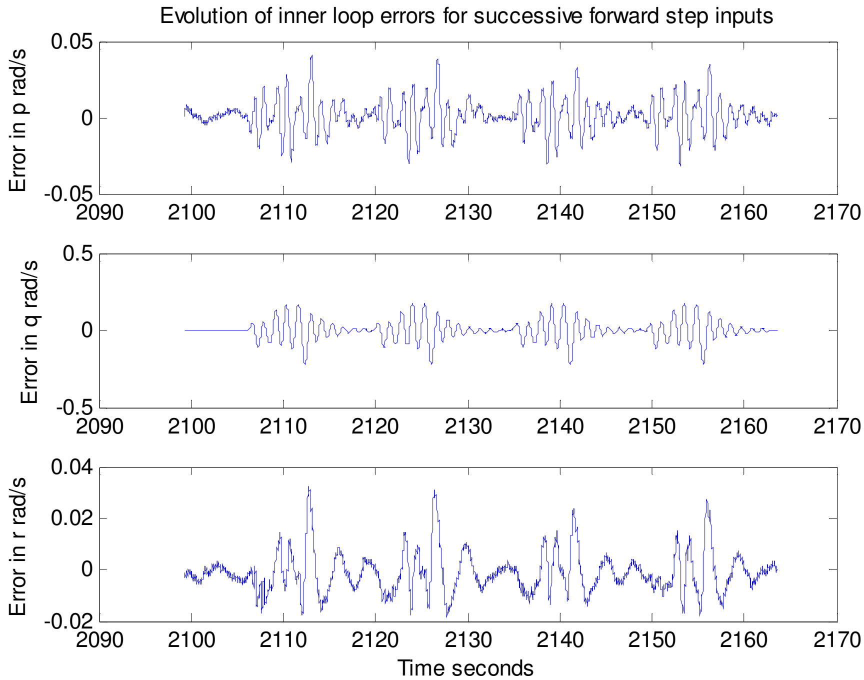

Figure 9 Evolution of inner loop errors for successive forward step inputs with only instantaneous adaptation

The forgetting nature of the controller is further characterized by the evolution of NN weights in the W and V matrices of equation 20. Figure 10 and Figure 11 clearly show that the NN weights do not converge to a constant value, in fact as the rotorcraft performs the successive step maneuvers the NN weights oscillate accordingly, clearly characterizing the instantaneous nature of the adaptation.

On the other hand, when both instantaneous and background learning NN learning law of *Method 2* is used a clear improvement in performance is seen characterized by the reduction in pitch rate error after the first two step inputs. Figure 12 shows the performance of the background learning augmented controller. The long term adaptation nature of the background learning augmented adaptive controller is further characterized by the convergence of NN weights in the W and V matrices of equation 20. Figure 13 and Figure 14 show that when background learning is used along with instantaneous learning the NN weights do not exhibit oscillations and tend to converge to constant values. This indicates that the NN learns faster and retains the learning even when there is a lack of persistent excitation. This indicates that the combined instantaneous learning and background learning controller will be able to perform better when performing a maneuver that it has previously performed, a clear indication of long term memory and semi-global learning.

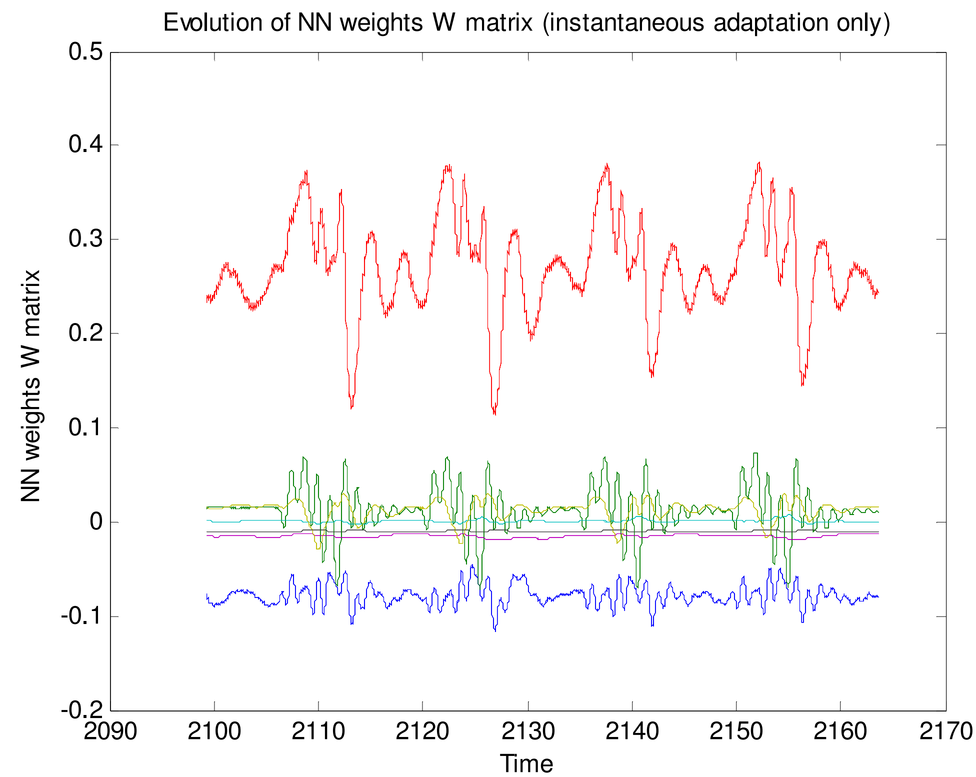

Figure 10 Evolution of NN weights, V matrix, with only instantaneous adaptation

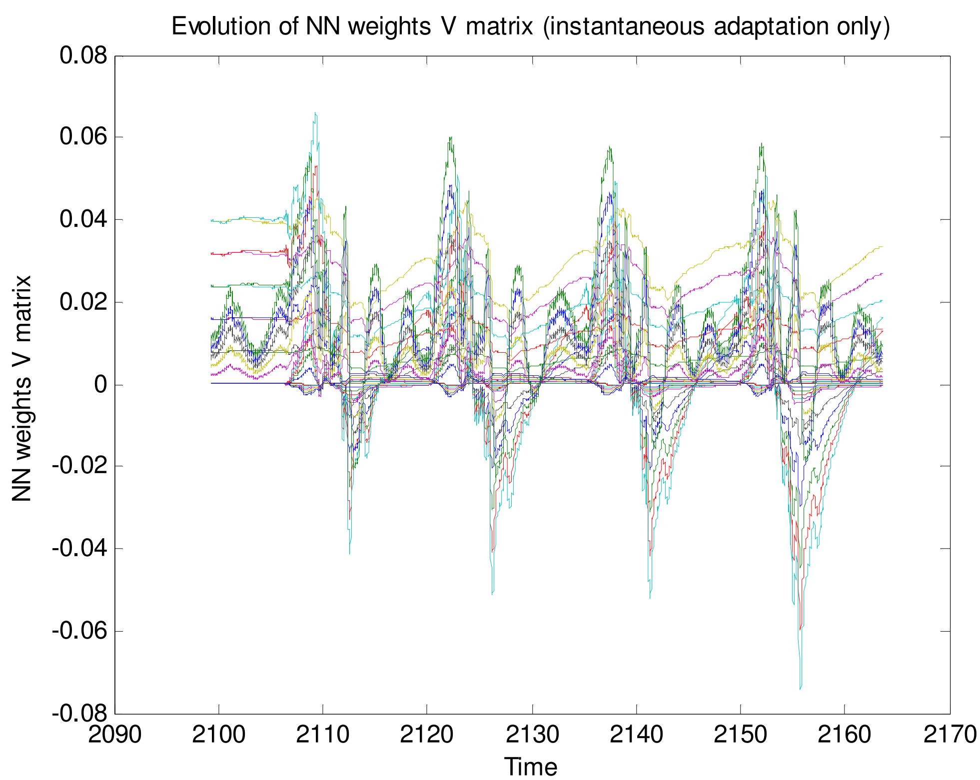

Figure 11 Evolution of NN weights, W matrix, with only instantaneous adaptation

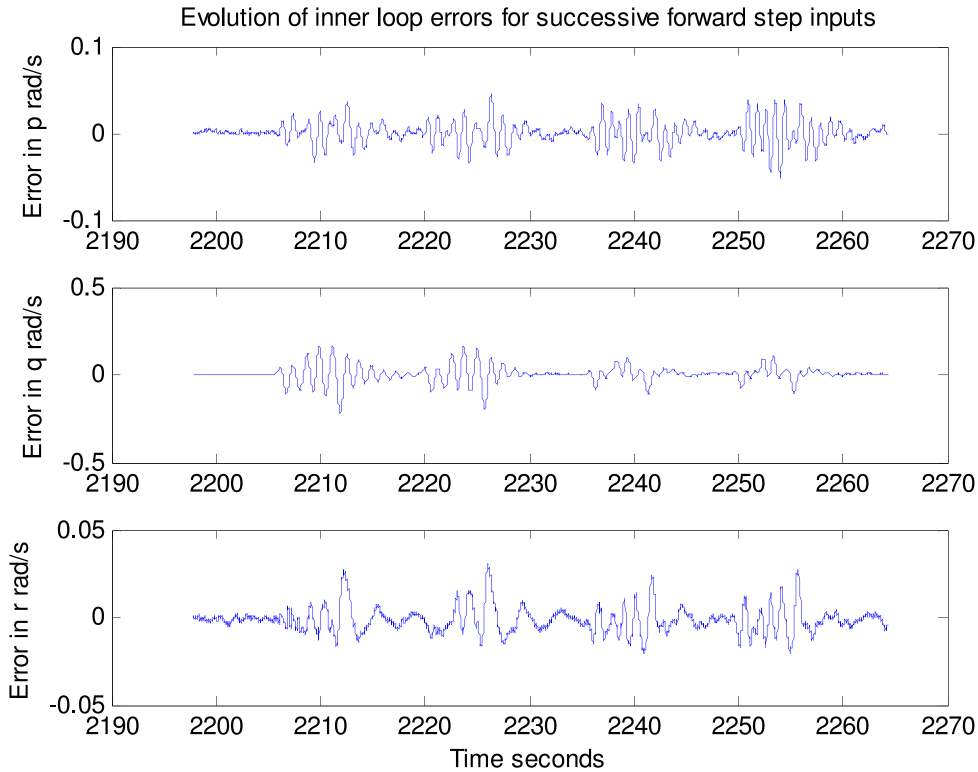

Figure 12 evolution of inner loop error with combined instantaneous and background learning controller

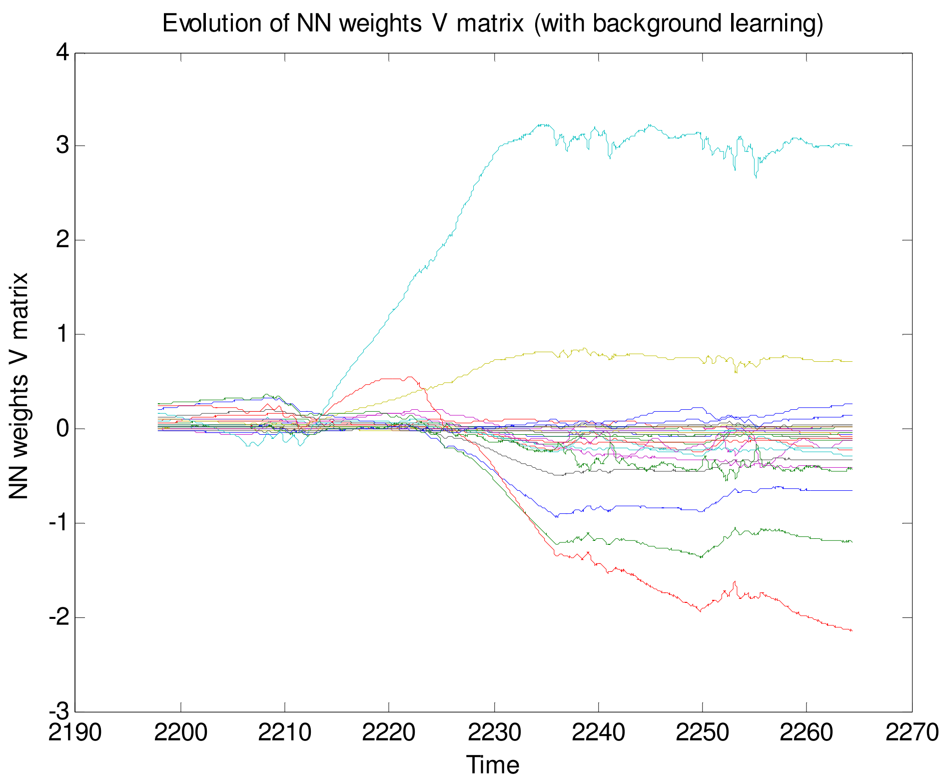

Figure 13 Evolution of NN weights, V matrix, with combined instantaneous and background learning

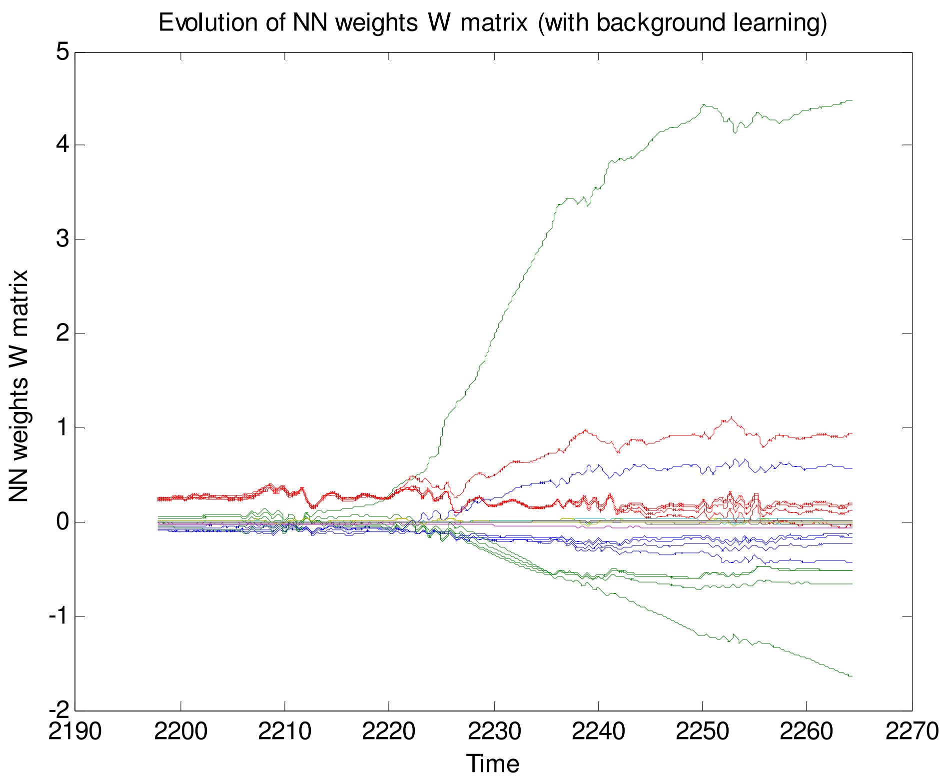

Figure 14 Evolution of NN weights, W matrix, with combined instantaneous and background learning

VII. Conclusion

We have proposed and a novel approach to the design of NN based controller which utilize online as well as background data. The new methods have the following advantages:

1. Long term learning: With combined online and background learning the adaptive law is able to retain long term learning. This allows the adaptive law to perform better when it encounters a task that it has adapted to before.
2. Semi Global adaptation: By carefully choosing background learning data points and storing them in a 'history stack' it is possible to maximize the dynamic envelop that the NN adaptive element is adapted to.
3. Increased Robustness: Since the network adaptation is dependent on more than one data point, it is less sensitive to occasional outlying signals.
4. Overcoming the Rank-1 limitation: The proposed adaptation law has higher rank than the unity, this results in better performance.

We have provided a proof of boundedness of all signals for a background learning LIP NN learning law. In the future we wish to expand the scope of this work by extending our theory to encompass various other training schemes and NN types. We also intend to incorporate robustness analysis and analyze the sensitivity of the new adaptation laws to the selection of the history stack data points.

Acknowledgments

The authors acknowledge the previous work of Seung Min Oh[8] with the second Author in this area. This work was supported in part by NSF #ECS-0238993.

References

[1] Spooner J.T., Maggiore, M., Ordonez,R. and Passino, K..M., *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximation Techniques*, John Wiley and Sons, 2002.

[2] Johnson, E., Kannan, S., "Adaptive Flight Control for an Autonomous Unmanned Helicopter", AIAA GNC 2002, Monterrey CA 2002.

[3] Kannan, S. K., and Johnson, E., N. Adaptive *Trajectory Based Control for Autonomous Helicopters*, AIAA Digital Avionics Conference, Oct 2002.

[4] Johnson, E. N. *Limited Authority Adaptive Flight Control*, Ph.D. Thesis, Georgia Institute of Technology, 2000.

[5] Kim, Byoung S., Calise, Anthony J., *Nonlinear Flight Control Using Neural Networks*, AIAA-1994-3646, August 1994.

[6] Kim, Y. H., Lewis, F.L., *High Level Feedback Control with Neural Networks, World Scientific Series in Robotics and Intelligent Systems Vol. 21*, World Science Publishing Co. Pte. Ltd. 1998.

[7] Lewis, F. L., *Nonlinear Network Structures for Feedback Control*, Asian Journal of Control, Vol.1, pp. 205-228, December 1999

[8] Johnson, Eric N, Oh, Seung-Min, *Adaptive control Using Combined Online and Background Learning Neural Network*, AIAA CDC 2004, USA.

[9] E. Johnson, M. Turbe and A. Wu, Georgia Institute of Technology, Atlanta, GA; S. Kannan, *Flight Results of Autonomous Fixed-Wing UAV Transitions to and from Stationary Hove,* AIAA-2006-6775, AIAA GNC 2006

[10] Hadad, W., Chellebaonia V., *Nonlinear Dynamical Systems and Control, A Lyapunov Based Approach*, Preprint, 2006, copyrighted by the Authors.

[11] Khalil H., *Nonlinear Systems,* Prentice Hall USA, 3 edition, December 18, 2001

[12] Gelb A., Applied Optimal Estimation, MIT press, 1974.